Deep Learning Fall 2024

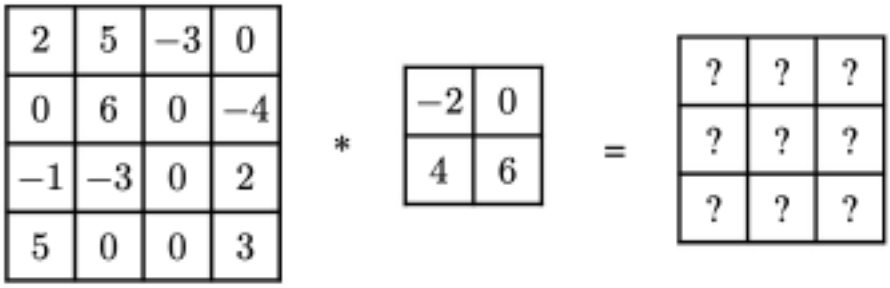
To do these exercises, you will use Python with 3 and the following packages:

- NumPy. This is often a good package to use, in order to create various data transformation / generate data.
- Pandas. With this package you can import your data into a data frame similar to how it's done in R.
- Matplotlib. This package allows you to graph your data and data transformations.
- PyTorch. This package is a versatile deep learning framework that facilitates building and training of neural networks.
- Seaborn. This package is built on top of Matplotlib, providing a high-level interface for creating informative and aesthetically appealing statistical graphics in Python.

You are not strictly forced to use these packages, but it is highly recommended. Feel free to use other packages you think are necessary.

Exercise 1 (Presence)

Explicitly calculated the output of the following convolution of a 4x4 input matrix with a 2x2 filter:



You can do the calculation by hand or write a custom program (e.g., with for loops) to carry out the calculation.

Exercise 2 (Take-home)

This exercise will extend on the model created during Exercise 1. You must now build a **convolutional** neural network that recognises numbers in pictures from the MNIST dataset.

Load the dataset by:

```
import torch
import torch.nn as nn
import torch.optim as optim
from torchvision import datasets, transforms
import seaborn as sns
import matplotlib.pyplot as plt

# Download the MNIST dataset
transform = transforms.ToTensor()
train_dataset = datasets.MNIST(root='./data', train=True,
download=True, transform=transform)
test_dataset = datasets.MNIST(root='./data', train=False,
download=True, transform=transform)
train_loader = torch.utils.data.DataLoader(dataset=train_dataset,
batch_size=64, shuffle=True)
test_loader = torch.utils.data.DataLoader(dataset=test_dataset,
batch_size=64, shuffle=False)
```

1. Dataset preprocessing:

a. Output the dimensions of train and test dataset and the size of the images. You can get access to the data by calling `.data` of the MNIST dataset (ie. `train_dataset.data`).
b. Plot the digit distribution using a bar histogram with the function `plt.bar()`.

2. Train a CNN:
    a. Design an architecture that contains the following:
        i. Convolutional Layer
        ii. MaxPool2D Layer
        iii. Fully Connected Layer
        iv. Dropout Layer
        v. ReLU Activation Function

    b. Achieve at least 95% accuracy in the test dataset by tweaking the architecture and save the model.

3. Check Results
    a. Plot the first convolutional layer filters before and after training the model.
    b. Feed the first convolutional layer an image and plot its activations.
    c. Plot some incorrectly labelled data points. Does it make sense why the network was not able to classify those data points?